

Network Level Defenses for Software-Defined Internet of Things

A thesis submitted for the Degree of Master of
Science in Computer Science

by

Bilal Ishfaq

Department of Computer Science and
Software Engineering University of Canterbury
New Zealand

2020

Contents

1	Introduction	8
1.1	Problem Statement	8
1.2	Research Questions and Objectives	10
1.3	Methodology	10
1.4	Motivation and Key Contribution	12
1.5	Thesis Structure	13
2	Related Work	14
2.1	SDN for IoT	14
2.2	Deception and MTD for IoT	17
2.3	Our Motivations and Rationale	20
3	Reactive Defense Mechanism	22
3.1	SD-IoT Network and Scenario Description	23
3.2	Proposed approach	24
3.3	System Model	25
3.4	Attacker Model	26
3.5	Defense Model	28
3.6	Metrics	32
3.7	Simulation	33
3.7.1	Simulation Settings	33
3.7.2	Simulation Steps	35

3.7.3	Simulation Results and Analysis	36
3.8	Results Discussion	39
3.9	Conclusion and Future Work	40
4	Proactive Defense Mechanism	41
4.1	Scenario Description	42
4.2	Methodology	44
4.2.1	Network Model	44
4.2.2	Attack Model	45
4.2.3	Defense Model	47
4.3	Security Failure Conditions	48
4.4	Metrics	49
4.4.1	Calculation of Metrics	50
4.5	Simulation	55
4.5.1	Simulation Setting	57
4.5.2	Network Scenarios to Calculate Metrics	58
4.5.3	Six schemes to compare the metric results based on when-to-shuffle and how-to-shuffle strategies	59
4.5.4	Comparative Analysis	60
4.6	Results Discussion and Conclusion	65
5	Conclusion	66
5.1	Limitations and Future Work	67

Acknowledgement

I am thankful to many people in making my Masters thesis possible. I would like to offer special thanks to my supervisors: Dr. Fabian Gilson, Dr. Dong Seong Kim and Dr. Mengmeng Ge for their invaluable guidance, supervision, and encouragement to me throughout this research. They not only provided me helpful suggestions, but also accepted responsibility to oversee this research, and guided me to the successful completion of this thesis. This thesis would not have been produced without their invaluable advice, excellent knowledge, unceasing support and enormous patience.

I would like to express my sincerest gratitude and appreciation to my parents for their endless love and support during my life. Without their moral support, this thesis would never have been completed. Last but not least, to all my sisters, their love and encouragements made this thesis possible. I would like to express my deep appreciation to my dear lab friends, who provided so much support and encouragement throughout this research and studies process. I wish them all the best in their future undertaking.

List of Abbreviations

ASLR Address Space Layout Randomization

AP Attack Path

APV Attack Path Variation

APE Attack Path Exposure

CVE Common Vulnerabilities and Exposures

DC Defense Cost

DDS Data Distribution Service

DDoS Distributed Denial of Service

DoS Denial of Service

FS Fix Shuffling

HARM Hierarchical Attack Representation Model

IDS Intrusion Detection System

IoT Internet of Things

LPWPANs Low Powered Wireless Personal Area Networks

MTTC Mean Time To Compromise

MTD Moving Target Defense

MTTSF Mean Time To Security Failure

NTS Network Topology Shuffling

NVD National Vulnerability Database

OS Operating System

RS Random Shuffling

SDN Software Defined Networking

SFC Security Failure Condition

VLANs Virtual Local Area Networks

WSNs Wireless Sensors Networks

Abstract

Internet of Things (IoT) has become a point of attraction to the industry and academia recently. The IoT is becoming popular because of the lesser prices and easy availability of smart devices. The IoT network consists of heterogeneous devices with limited computational and power resources, and not equipped dynamically to respond to abnormalities. Besides, some of the IoT nodes in an IoT network are attached with vulnerabilities. To ensure the security for such types of IoT networks, we propose a reactive defense mechanism and integrated proactive defense mechanism of an software-defined IoT (SD-IoT) network. The reactive defense mechanism provides the maximum number of hard to exploit vulnerable IoT nodes along the path to the base station, after reconfiguration. To reconfigure the IoT network topology, we develop and implement a reconfiguration algorithm, using the software-defined networking (SDN) controller. The algorithm reconfigures the topology, when the intrusion is detected in the IoT network.

In the integrated proactive defense mechanism, we use cyber deception along with the Moving Target Defense (MTD). The cyber deception includes the decoy system and attracts the attacker towards itself. On the interaction of the attacker, the decoy system captures the intentions of the attacker. The MTD makes the attack surface hard by shuffling the connections between IoT nodes. To analyze the effectiveness of our proposed defense mechanisms, we measure the security metrics using the Hierarchical Attack Representation Model (HARM).

The results of our reactive defense mechanism and integrated proactive defense mechanism show the increase in attack efforts.

In summary, the contributions of the thesis are; 1) To develop a reactive defense mechanism and reconfiguration algorithm to changes the IoT network topology on intrusion detection and to calculate the mean-time-to-compromise (MTTC) security metric that shows the effectiveness of our defense mechanism; 2) To develop an integrated proactive defense mechanism, implement the cyberdeception and MTD as defense strategies, develop, implement and calculate the security metrics (i.e., Attack Path Variation, Attack Path Exposure, mean-time-to-security-failure and Defense cost) to show the effectiveness of our defense mechanism.

Chapter 1

1 Introduction

It has been observed that the Internet of Things (IoT) is becoming an essential part of human being lives for the last decade [1]. IoT provides a transition of real things into virtual where all objects can be readable, addressable, and discoverable over the internet [2]. The IoT networks consist of IoT nodes and the IoT nodes communicates with each other via different communication protocols such as wireless standard Bluetooth [3], WiFi [4] and ZigBee [5]. These IoT devices are deployed at different locations, e.g., bus stops, street lights or underground in the water, to collect required data. The IoT devices have less computation and power storage capacity.

1.1 Problem Statement

An IoT network consists of heterogeneous IoT devices. These heterogeneous IoT devices are attached to different exploitable vulnerabilities that make IoT networks vulnerable to attackers. These vulnerabilities can be classified as known and unknown vulnerabilities shown in Figure 1. The known vulnerabilities attached to the IoT nodes could be patchable or non-patchable (forever day) vulnerabilities. Attackers can exploit these vulnerabilities to launch attacks such as Distributed Denial of Service (DDoS) attacks [6], node capture and control [7], physical damage [8] and reconnaissance [9]–[11] at-

tacks. It is one of the objectives to mitigate the impact of such attacks on IoT devices and to secure IoT networks. The ideal solution to mitigate the

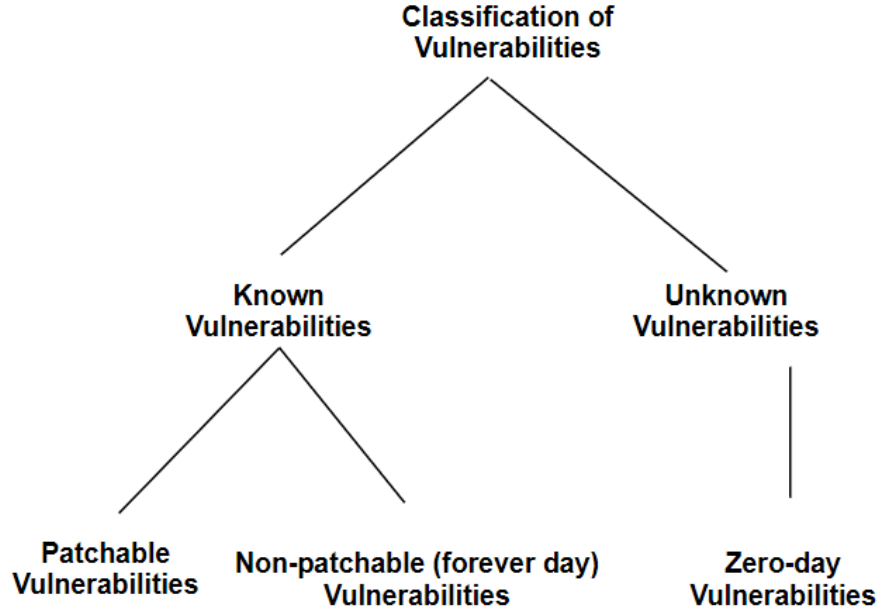


Figure 1: Scenario Description

attack impact is to remove all the patchable, non-patchable and zero-day vulnerabilities associated with the IoT devices. However, it is not achievable because of the time and cost constraints. Besides, the zero-day vulnerabilities (not known to the vendor and identified by the attacker) are always exploitable by attackers. There is a need for alternative defense mechanisms to respond to attacks and to secure the IoT networks in real-time.

1.2 Research Questions and Objectives

Q1: What defense mechanisms can we use proactively to harden an attack surface of an IoT network?

Q2: What defense mechanism can we use reactively to make the attack surface hard and to respond to the intrusions detected, in an IoT network?

The objective of our work is to protect the SD-IoT network from attackers by applying integrated proactive defense mechanism (to address Q1) and reactive defense mechanisms (to address Q2). The two objectives to address the research questions are as follows.

- 1) To make the attack surface hard using an integrated proactive defense mechanism.
- 2) To reconfigure the SD-IoT network topology using a reactive defense mechanism.

1.3 Methodology

We follow the same methodology for each of our proposed defense solution. Our methodology consists of the following scenarios.

System Model: In our system model, we consider the SD-IoT network and analyze it based on real-world network scenarios such as smart hospitals, smart offices, and smart homes, etc. We explain the type of IoT devices used in the network along with the network scenario. We assume the size of the network and area of deployment (i.e., open environment or in a building

based on hypotheses). Finally, we explain how communication takes place between the IoT nodes in the IoT network.

Attacker Model: In the attacker model, we assume the attacker's privileges to compromise the node and intentions to compromise the IoT network. The assumptions also include how attacker select the target and how he/she exploit the different paths to reach the target. The attacker model also explains the knowledge of the attacker, he/she already have about the IoT network such as the exploitable vulnerabilities information and the entry point to the IoT network. The attacker uses this knowledge to find the entry point and to compromise the SD-IoT network.

Defense Model: Our defense model includes the defense solutions to counter the attacks and to protect the IoT network. The defense solutions are the different techniques or mechanisms which are used to protect the SD-IoT network. The defense solutions are implemented via SDN controller to the IoT network. The SDN controller changes the network topology at defined security failure conditions. We implement the defense solutions at the network level or at the node level and evaluate it based on the defined system model and attacker model as explained above.

Evaluation: We conduct simulations for evaluation. The simulation validates our proposed defense techniques via security metrics. We use the simulations scenario based on our proposed system model and the attacker model.

1.4 Motivation and Key Contribution

The motivation behind our work is to provide security solutions for IoT networks consist of easy to exploit vulnerability of an IoT node and hard to exploit vulnerability of an IoT node. The attacker exploits the vulnerability to compromise the IoT node and the IoT network. To protect the IoT network from the attacker, we contributed to the IoT security-related research field and propose two defense mechanisms, namely a proactive defense mechanism and a reactive defense mechanism. The proactive defense mechanism includes the cyberdeception and MTD techniques to make the attack surface hard for the attacker. The reactive defense mechanism includes the IoT network topology reconfiguration on intrusion detection to make the attack surface hard for the attacker.

For the security analysis of the IoT network, we use the Hierarchical Attack Representation Model (HARM) [12] and measure the security metrics. The HARM takes the topology information as an input and constructs two layers. The Upper layer includes the attack graphs for the attack paths from the entry point up to the base station. The lower layer includes the vulnerability information of the nodes.

To address the research question Q1, our contribution is as follows:

- Propose a proactive defense mechanism that uses a concept of deception and MTD techniques.
- Design different network scenarios and use shuffling schemes based on

deception and MTD techniques.

- Use the metrics to measure the effectiveness of the proposed proactive defense mechanism.

To address the research question Q2, our contribution is as follows:

- Propose a reactive defense mechanism for SD-IoT networks that consists of easy to exploit vulnerability and hard to exploit vulnerability of IoT nodes.
- Develop the algorithm to reconfigure the IoT network topology that gives the maximum number of hard to exploit vulnerability of IoT nodes along the path to the target.

1.5 Thesis Structure

The rest of the thesis is organized as follows. Chapter 2 reviews the literature and summarizes techniques for IoT network security. Chapter 3 presents the reactive defense mechanism for SD-IoT networks. Chapter 4 explains the integrated proactive defense mechanism for SD-IoT networks. Finally, Chapter 5 concludes the thesis.

Chapter 2

2 Related Work

SDN makes network management easy compared to traditional IP networks [13]. SDN gives centralized control by separating the control plane from the data plane and manages all the devices with a single controller [14]. In the following, we discuss the solutions of SDN for the IoT networks to explore the feasibility of IoT integration with SDN and present the work addressing the SD-IoT security issues.

2.1 SDN for IoT

Liu et al. [15] proposed an SDN based IoT architecture for smart urban sensing. They carried a quantitative analysis and used a case study to present the benefits of SD-IoT architecture. The proposed architecture incorporates three layers. The physical layer consists of servers, forwarding devices, sensor platforms, and gateways. It collects the data from the smart environment and sends it to the server. The control layer is made up of a cloud controller, network controller, and the sensor controller to manage the devices of the physical layer through southbound APIs. This research work proposes the solution for integrating SDN with an IoT architecture and confirms the feasibility of SD-IoT architecture.

Hakiri et al. [16] proposed an IoT architecture that integrates a data

distribution service (DDS) and SDN. The DDS is a message-oriented publish/subscribe and acts as middleware. The architecture includes the IoT gateways which are SDN-enabled and, are connected to the smart devices. The IoT gateways are also connected to the SDN forwarding devices, that forward the packets according to defined flow rules, via southbound interfaces. This research addresses the mobility and the scalability issues in the SDN-enabled IoT gateways framework that ensures the integration of SDN with IoT. Sandor et al. [17] developed an SDN based algorithm for hybrid IoT network architecture. The hybrid architecture includes non-SDN topology segments (i.e., non SDN switches) and SDN switches with redundant communication points. The redundant communication points were induced to enhance the resilience of the network and to maintain the performance continuous in the presence of the attack. The algorithm is used to evaluate the performance of reconfiguration by switching the path from the attacked path to the redundant entry points, on DDoS attack detection. However, the proposed solution is semi-SDN based. Besides, the redundant paths do not change the attack surface as the position of the redundant communication points is the same. It only gives the alternative way for the traffic. Our proposed approach is fully SDN-based and switch the connections between the IoT nodes to change the attack surface.

Grigoryan et al. [18] proposed an SDN based IoT architecture that includes IP based cooperative security for multiple IoT networks. The SDN controller first identifies the malicious activities provided by the end sys-

tem (e.g., end users that is the entry point of attacker) and then broadcast these malicious activities to the network level in multiple network environments, to stop the malicious traffic. Also, the SDN controller in the proposed architecture blocks potential attackers and evaluate the authenticity of the architecture itself by double-checking the traces of malicious traffic. The proposed work gives the IP based security solution for SD-IoT networks. Our approach is based on MTD technique that changes the attack surface by shuffling the connection between IoT nodes. Besides, our approach is better as the attacker loses all the gathered information about the entry points on reconfiguration.

Chakrabarty et al. [19] proposed a protocol for secure communication in the IoT network using SDN. They proposed a solution for encryption of both payload and meta-data in the network layer and link layer. The centralized SDN controller works as a reliable third party and uses black packets [20] to communicate with the IoT resource-constrained devices. They carried out simulations which use different modes of the nodes (awake mode or asleep mode) and different topologies. The results showed enhanced security provided by the proposed architecture than the existing 802.15.4 protocol.

In this section, we investigated related works proposing solutions to enhance the security of SD-IoT networks. We found it feasible that SDN can manage different types of devices, i.e., SDN with wireless sensor networks (WSNs) as sensing devices, SDN with Mobile networks as user devices, and SDN with IoT as a smart environment. The SDN controller is programmable

and uses this functionality to configure all the IoT devices. It makes the administration tasks easy compared to the traditional network, and the network administrator can define the flow rules and view the IoT network from a single central SDN controller.

2.2 Deception and MTD for IoT

Deception is the defensive technique that is in addition to the traditional defense mechanisms (e.g., Intrusion Detection System (IDS), Firewall or anti-software, etc) [21]. La et al. [22] used the honeypot-enabled decoy IoT network and proposed a game theoretic model. In the model, the attacker interacts with the defender and deceive the defender via suspicious or seemingly traffic. As a defense, the defender uses a honeypot system to capture the attacker. Anirudh et al. [23] proposed a honeypot (decoy) model for online servers in IoT network to mitigate the DoS attack. They use the decoy server in the main server and mitigate the DoS attack in the decoy server. The mitigation of the DoS attack at the decoy server prevents the complete shutdown of the IoT network. However, the work [22], [23], as explained above have not analyzed the deception impact on network level security for SD-IoT network.

MTD is a defense technique applied to a network to continuously change the attack surface for the attacker by network reconfiguration [24]. MTD is categorized into three classes namely Shuffling, diversity, and redundancy [25]. The shuffling MTD technique changes the network configuration to

confuse the attackers. The configuration changes could be network addresses or network topology reconfiguration. The diversity MTD techniques make the attack surface hard by using different system components (e.g., software with different kinds of operating systems while provide the same functionality) with the same functionality. The redundancy MTD techniques change the attack surface dynamically via replicas of the network components [26].

Ge et al. [27] evaluated the performance of the existing MTD technique called Address Space Layout Randomization (ASLR) using the HARM. This MTD technique is deployed for the IoT nodes that makes difficult for the attacker to find the locations of IoT nodes in randomly placed areas. Hence it increases security by widening the search space area. They used the attack success probability, compromise rate, and attack cost as metrics to evaluate this technique. This defense mechanism only uses MTD while our proposed defense mechanism uses decoy system with MTD. The decoy system monitors the attacker intentions when they interact with it. Also, we implement and measure additional security metrics such as attack path variation (APV) and attack path exposure (APE) to randomize the attack paths.

Sherburne et al. [28] proposed an IP shuffling based MTD technique. The IP addresses of IoT devices change dynamically using the concept of a random assignment approach. They used the Low Powered Wireless Personal Area Networks (LPWPANs) as a protocol for IoT devices communication. Zeitz et al. [29] extended the approach proposed by Sherburne et al. [28]. They presented the design for the MTD technique that is based on IP address

rotation while our defense mechanism is based on change in connections between nodes.

Ge et al. [30] introduced the proactive defense mechanism for the SD-IoT network. They developed two proactive defense mechanisms to reconfigure the SD-IoT network. In these two proactive defense mechanisms, they considered two cases: The SD-IoT network that includes patchable and non-patchable nodes, and the SD-IoT network only includes non-patchable nodes. They conducted the simulation and measured the security metrics to evaluate their defense mechanism. They used the optimal method and the heuristic method to reconfigure the SD-IoT network topology. In the Optimal method, the algorithm introduced a maximum number of patchable nodes in the path from the entry point to the target. However, this algorithm has limitations of hops restriction. The algorithm cannot change (increase or decrease) the hop count for the nodes after reconfiguration. The heuristic method eliminated the limitation. However, the heuristic method did not guarantee the maximum number of patchable nodes or the hard to exploit node in the path to the base station. In our work, we proposed the algorithm that offers the maximum number of hard to exploit IoT nodes along the path to the base station.

Kouachi et al. [31] uses a MTD technique for packet flow anonymization in IoT network. They addressed the security issues related to the identification of communication flow and tracking of the packets. They proposed “micro One Time Address” as a defense solution for power constraint (less power

storage capacity) IoT devices that enhance the life of the IoT devices . This defense solution changes the structure of the IPv4 packets and uses the single IP address (instead of using all the information in the packet header), used to transmit the packet, for verification. However, the solution causes reconfiguration overhead and the change of IP header requires the reconfiguration of all involved routers.

Nizzi et al. [32] proposed a lightweight MTD technique that shuffles IP addresses. The MTD technique is named HMAC (AShA) which is based on a hash function. It changes the IP address in the IoT network and permits the IoT devices to recompute their addresses when a multicast message is sent to the network by the switch. This MTD technique is based on IP address shuffling to counter the attack while our MTD technique is based on changing the connections to stop the attacker.

2.3 Our Motivations and Rationale

We explained the different defense mechanisms in the literature review, used for SD-IoT networks to protect the network from attackers. The exceptions to this are the security solutions given by [17], [30].

However, our proposed work includes a reactive defense mechanism that changes the connections at the edges, from easy to exploit vulnerability of IoT nodes to hard to exploit vulnerability of IoT nodes during topology reconfiguration and removes the hop count limitation of as in [30].

Our proposed work also includes an integrated proactive defense mecha-

nism that uses cyberdeception with MTD to make the attack surface hard for the attacker. The decoy system includes the decoy nodes and captures the intentions of the attacker when the attacker interacts with the decoy nodes. The MTD changes the attack surface dynamically by reconfiguring the SD-IoT network topology. On reconfiguration, the attacker loses the gathered information about the network.

Chapter 3

3 Reactive Defense Mechanism

The motivation behind our work is to change the attack surface that lowers the probability of successful attacks. To achieve this goal, we provide a defense mechanism for IoT networks that consist of easy to exploit and hard to exploit vulnerabilities of IoT nodes. We aim to offer the maximum number of IoT nodes, having hard to exploit vulnerabilities, to the attacker along the path to the target. Therefore we propose a defense mechanism named reactive defense mechanism. It reconfigures the IoT network topology on intrusion detection and replaces the IoT nodes having easy-to-exploit vulnerabilities with the IoT nodes having hard-to-exploit vulnerabilities along the path from the entry point of the attacker up to the target.

To reconfigure the IoT network topology, we use a reconfiguration algorithm. For the security analysis of the IoT network, we use HARM [12]. It constructs the attack paths from the entry point of the attacker up to the target of the attacker, and we compute MTTC across each attack path. The answer to the research question Q2 “How can we respond to the intrusions reactively, for IoT network” is given by the following contributions.

- Propose a reactive defense mechanism for the SD-IoT network.
- Develop an algorithm to reconfigure the IoT network topology on in-

trusion detection.

3.1 SD-IoT Network and Scenario Description

The proposed IoT network as shown in Figure 2 follows a tree topology and consists of heterogeneous IoT nodes having easy to exploit vulnerabilities and hard to exploit vulnerabilities. The attacker can exploit the vulnerabilities attached with the IoT nodes and can break into the IoT network. Once the attacker successfully exploits the first (entry point) IoT node, he/she tries to exploit the neighbor IoT nodes of the entry point node to reach the base station. On intrusion detection by the IDS, the SDN controller reconfigures the network topology to block the attackers.

We consider a smart environment of the IoT network as a scenario that consists of two types of IoT nodes as follows.

- Noise sensor IoT nodes
- Weather sensor IoT nodes

The noise sensor IoT nodes have easy to exploit vulnerabilities and weather sensor IoT nodes have hard to exploit vulnerabilities. As shown in Figure 2, the red (squares) IoT nodes are noise sensors and the blue (circles) IoT nodes are weather sensor IoT nodes respectively.

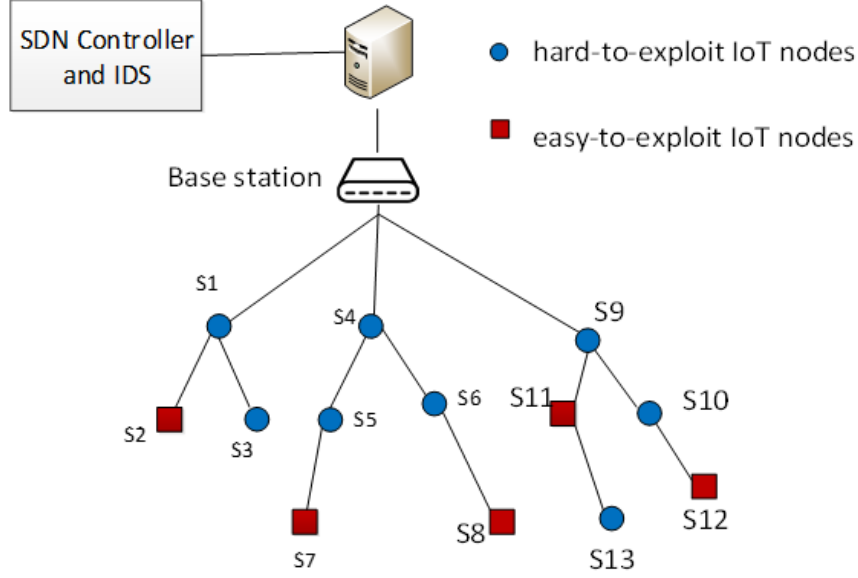


Figure 2: An Example Scenario and Configuration

3.2 Proposed approach

The IoT network as discussed in Section 3.1 consists of different IoT nodes. To protect the IoT network from attackers, we change the attack surface by reconfiguring the IoT network topology using a reactive defense mechanism. After reconfiguration, we obtain a reconfigured IoT network topology which offers the maximum number of IoT nodes having hard to exploit vulnerabilities from the entry point of the attacker up to the target (base station). The topology reconfiguration takes place on intrusion detection by IDS. We make the assumption to compute all possible sets of topologies, offering the maximum number of IoT nodes having hard to exploit vulnerabilities along the path to the base station. These pre-calculated sets of topologies give the

same number of hop counts (connection link between two neighbor nodes is one hop) in the initial topology. On intrusion detection, the algorithm applies suitable pre-calculated topology that offers the maximum number of IoT nodes with hard to exploit vulnerabilities.

To summarize, the reactive defense mechanism limits the topology changes by pre-calculating the set of topologies before intrusion detection and applies one of the pre-calculated topology on intrusion detection.

3.3 System Model

As shown in Figure 2 in Section 3.1, the IoT network consists of an IDS, an SDN controller, a base station (single board computer with 2.4 GHz RF transceiver), and IoT nodes. The base station acts as a gateway between the SDN controller and the IoT nodes and transfers the received data by the IoT nodes to the SDN controller. We consider the Snort IDS [33].

In this setting, the base station receives data from the IoT nodes and sends it to the SDN controller for query (sent by the user, i.e., weather query or noise query) processing. The IoT nodes communicate via the ZigBee communication protocol [5]. This protocol works on low bandwidth, low latency, and low energy consumption that is ideal for resource constraint nodes. However, in terms of security, ZigBee protocol becomes vulnerable for multi-hop communication and relies on network security. If the attacker successfully exploits the node, it becomes easy to compromise the ZigBee protocol and the attacker can reach the target.

3.4 Attacker Model

In the attacker model, we consider that the attacker scans for the IoT nodes in the IoT network. The attacker scans for multiple vulnerable IoT nodes and chooses one of them as an entry point. After choosing a node, the attacker tries to compromise it. In the meantime, the IDS detects the attacker's presence (we consider 100 percent detection accuracy of IDS) and the reconfiguration algorithm changes the position of the attacked node. As a result, the attacker loses the connection and try to connect with another vulnerable IoT node. For the vulnerability analysis, we use the Common Vulnerabilities and Exposures (CVE) / National Vulnerability Database (NVD) [30] for IoT networks. The vulnerabilities of the nodes, vulnerabilities score, MTTC for each vulnerability and the impact are listed in Table 1. For each IoT node, we choose a random vulnerability value from the given table. We use the uniform random function to choose and assign the vulnerability value for each IoT node.

We have defined a certain threshold value (0.5 as it has medium impact and act as a border between low impact and high impact) based on vulnerability score which specifies whether the IoT node has easy to exploit vulnerability (CVSS-BS ≥ 0.5) or hard to exploit vulnerability (CVSS-BS < 0.5). The calculation of the MTTC for the vulnerability is explained in Section 3.7.3.

In our attacker model, we assume an attacker with the following assumptions:

Table 1: Vulnerability Score and MTTC Values

Vulnerability Details	CVSS Score	MTTC Value (hrs)	Impact
CVE-2019-2338	0.71	1.39	High
CVE-2019-2322	0.98	1.05	High
CVE-2019-2318	0.55	1.78	High
CVE-2019-10616	0.45	2.30	Low
CVE-2019-2261	0.45	2.30	Low
CVE-2019-2315	0.78	1.21	High
CVE-2019-10591	0.75	1.33	High
CVE-2019-10590	0.98	1.05	High
CVE-2019-10587	0.98	1.05	High
CVE-2018-11976	0.45	2.30	Low
CVE-2017-8329	0.46	2.11	Low
CVE-2019-11820	0.45	2.30	Low
CVE-2019-14071	0.78	1.21	High
CVE-2017-11578	0.49	2.01	Low

- The entry point to the IoT network is an IoT node.
- The attacker scans the IoT network using reconnaissance attack and exploit the vulnerability associated with the node to compromise the node.
- When a node is compromised, the attacker becomes part of the IoT network.
- The attacker compromises all nodes between the entry point and the base station.
- Once the attacker reaches the base station then he/she can compromise the entire network.

3.5 Defense Model

In our defense model, we propose an algorithm to reconfigure the IoT network topology on intrusion detection. Our objective is to make the attack surface hard by offering maximum number of IoT nodes having hard to exploit vulnerabilities from the entry point of the attacker to the base station. We implement Algorithm 1 for topology reconfiguration and use the following notation to express the algorithm.

- s_i : The IoT nodes in the network ($i \in \{1, \dots, n\}$). where n is the total number of IoT nodes.
- b_{s_i} : one neighbour IoT node.
- R_T : The initial IoT network topology before reconfiguration.
- R_f : The IoT network topology after reconfiguration (when the intrusion is detected).
- v_{s_i} : A vulnerability value of s_i .
- P_{s_i} : The current parent node of s_i .
- h_j : A hop count calculated between two IoT nodes in an IoT network, e.g., entry point node to its parent node and then the parent of the entry point node's parent and so on.
- H_i : Total number of hops in R_f , in one path (i.e., the number of hops in one path of the optimal topology after reconfiguration).

- H_{max} : The maximum number of hops in one path of RT , i.e., the number of hops in one path of the initial topology.
- $v_{b_{s_i}}$: Vulnerability value of neighbour (b_{s_i}) IoT node.
- N : Total number of IoT nodes with hard to exploit Vulnerabilities.
- X_i : A single IoT node with hard to exploit vulnerability.
- K_i : One attack path from entry point node up to the target. i.e., it is a set of nodes.
- C_{s_i} : The list of elements in a communication range of S_i .
- K_{RF} : An optimal path which gives maximum number of IoT nodes having hard to exploit vulnerabilities and minimum number of hops after reconfiguration.
- T : Threshold value based on CVSS-BS to check the easy to exploit vulnerability and hard to exploit vulnerability of IoT node.
- A^* : To initialize variables.

The algorithm reconfigures the IoT network topology according to the following steps:

- Compute the possible reconfigured set of topologies in the same communication range of IoT nodes before intrusion detection and apply the suitable pre-computed topology on intrusion detection.

Algorithm 1: Topology Reconfiguration

```

1 for each  $b_{s_i} \in C_{s_i}$  do
2    $K_{R_F} \leftarrow A*$ ;
3    $H_i \leftarrow A*$ ;
4    $N \leftarrow A*$ ;
5   while  $b_{s_i}$  belongs  $P_{s_i} \neq \Phi$  do
6     Add  $b_{s_i}$  into  $K_i$ ;
7     if  $v_{b_{s_i}} < T$  then
8       | Add  $X_i$  into  $N$ ;
9     end
10    Add  $h_j$  into  $H_i$ ;
11     $b_{s_i} \leftarrow b_{s_i} [p_{s_i}]$ ;
12  end
13  if  $H_i = H_{max}$  then
14    | Add  $N$  and  $H_i$  into  $K_{R_F}$ 
15  end
16  Return  $k_{R_F}$ 
17 end

```

- If the node at the entry point of the attacker have easy to exploit vulnerability, it checks for the next neighbor IoT nodes, whether it is easy to exploit vulnerable or hard to exploit vulnerable.
- It establishes the connection with the IoT node which have hard to exploit vulnerability and offer the same number of hops from that IoT node to the base station.

The explanation of algorithm is as follows.

In line 1, the algorithm checks the communication range for a node to its neighbor nodes, and gives the list of nodes in the same communication range. Line 2 initiates the path list that is empty at the initial stage and after that includes the path based on the maximum number of IoT nodes with hard to exploit vulnerabilities. Line 3 and 4 initialize the hops and number of IoT nodes having hard to exploit vulnerabilities respectively. Line 5 is the condition which picks the nearest neighbor IoT node with hard to exploit vulnerability in the communication range, of the IoT node that is entry point for the attacker. Then algorithm checks the parent IoT node of that neighbor IoT node and check if it is in the path list of elements in the communication range. If it is not in the path list, the loop breaks for current IoT node and starts for the next IoT node in the path. The loop keeps checking for all the IoT nodes until the base station (root node). Line 6 adds all the neighbor IoT nodes of the IoT node that is under scanning attack and gives the path list from entry point till base station. Line 7 includes the vulnerability value

of current neighbor IoT node and compares it with the threshold value. If the vulnerability value of the neighbor IoT node is less than the threshold value, the IoT node is considered hard to exploit vulnerable and added it to the hard to exploit variable in line 8. Line 10 adds each hop of the path into the total number of hops, each time the while condition becomes true. Line 11 includes the next neighbor IoT node which the while condition checks at line 5, to identify if it is already included in the path list or not. If it is not in the path list then the while condition executes again up to line 11. Line 12 ends the while condition. Line 13 compares the initial topology hops with the hops counted during reconfiguration. If the counted hops are equal or less to the initial topology hops, we consider it as one path. This path is added to the total number of paths offering maximum number of IoT nodes with hard to exploit vulnerabilities, in line 14. Line 15 ends the condition started at line 13. Line 16 returns the total number of paths. Line 17 ends the loop, started at line 1.

3.6 Metrics

To evaluate our defense mechanism, we use the following metrics.

- Number of IoT nodes with hard to exploit vulnerabilities: Maximum number of hard to exploit vulnerabilities of IoT nodes in the attack path represents the optimal attack path.
- MTTC: It is the time (hour) to compromise all the IoT nodes by the

attacker from the entry point upto the base station.

The MTTC is the time, the attacker takes to compromise all the IoT nodes in the path, from entry point up to the base station. When the IDS detects an intrusion, it triggers the reconfiguration of the IoT network topology. After reconfiguration, the number of IoT nodes with hard to exploit vulnerabilities are more along the path to the base station and the attacker takes more time to compromise the IoT nodes. There are less chances for the attacker to exploit all the IoT nodes and less chances to reach the base station.

3.7 Simulation

In our simulation, we consider the IoT network based on Figure 2. We consider the following machine specifications and the IoT nodes. We use the desktop machine with the operating system, Windows 10 64-bit, 8.0 GB RAM, and Core i5-8250u (1.60GHz) processor. We consider the IoT nodes are based on microcontrollers and configured with the ZigBee radio module based on IEEE 802.15.4 standard [5]. To achieve the programmability via the SDN controller, we consider SDN-WISE protocol [34].

3.7.1 Simulation Settings

In simulation settings, we use a SD-IoT network consisting of 100 IoT nodes. In the network, 50 percent of the IoT nodes have easy to exploit vulnerabilities and 50 percent of the IoT nodes have hard to exploit vulnerabilities, to

investigate the performance of our implemented algorithm. We assume the presence of an attacker in the IoT network and the attacker's main goal is to compromise the base station. We assume that the IoT nodes are installed in a specific area of interest which is a smart office in our case, and send the gathered data (weather update, noise) from the surroundings towards the base station. We distribute the IoT nodes in a circle form and define the simulation area that is 360 meters * 360 meters [30]. The average distance between the two IoT nodes is 25 meters and they communicate with each other via ZigBee protocol. The ZigBee [5] protocol specifications indicate the transmission range of ZigBee, which is 10 meters to 100 meters depending on the power output of the individual device and the surrounding environment. We use the 75 meters of average communication range in our simulations (e.g., one node can communicate with any of the other node within this range).

The SDN controller defines the routing policies for the packets flow at base station. The communication between the IoT nodes takes place via steps explained in defense model of Section 3.5. The attacker chooses the entry point node randomly and try to reach the base station by following the steps as discussed in the Section 3.4. When the attacker scan for the IoT node, the intrusion is detected by the IDS. It triggers the reconfiguration algorithm and the connections shuffling takes place between the nodes. The attacker loses the connection and try to exploit another node to break into the IoT network. To evaluate our defense mechanism, we compute the MTTC

[30] before and after reconfiguration, across each path from the entry point. We calculate the number of IoT nodes with hard to exploit vulnerabilities along the path in the reconfigured topology and use the HARM to compute all the possible attack paths from entry point up to the base station. The percentage of easy to exploit and hard to exploit vulnerabilities of IoT nodes is equal in both topologies (e.g., before and after reconfiguration).

3.7.2 Simulation Steps

We compute the MTTC before reconfiguration and after reconfiguration of IoT network topology. We used the reconfiguration algorithm to reconfigure the randomly generated IoT network topology. On reconfiguration, we have the reconfigured IoT network topology and compute all the possible paths that attacker uses to reach the base station. We calculate the MTTC of the paths for initial IoT network topology and reconfigured IoT network topology.

To compute the MTTC, we consider the following steps:

- We consider the presence of an attacker in the IoT network.
- The attacker chooses the entry point randomly.
- On the detection by IDS, the SDN controller reconfigures the topology, and the attacker loses the current connection.
- The attackers choose another entry point to break into the IoT network

and continue to compromise the IoT nodes until they reach the base station.

Initially, the detection takes place only once to trigger the topology reconfiguration. The reason to trigger the topology reconfiguration is to measure the MTTC for the path with the maximum number of hard to exploit vulnerable IoT nodes. Because, we get the topology with the maximum number of hard to exploit vulnerable IoT nodes after reconfiguration.

3.7.3 Simulation Results and Analysis

We use the example IoT network as shown in Figure 2 for our simulation. We perform the following steps in our simulation.

- We generate the IoT network 10 times with an equal proportion of easy to exploit and hard to exploit vulnerable IoT nodes.
- We increase the size of the IoT network by 10 IoT nodes each time.
- We randomly assign the vulnerabilities to the IoT nodes.
- We run the simulation 100 times for each time the IoT network generated.
- We regenerate a new IoT network topology for each run for a given size.
- The attacker randomly select the entry point IoT node, each time the IoT network is regenerated.

We use the reconfiguration algorithm as explained in Section 3.5, to reconfigure the IoT network topology. It gives all possible paths from the entry point to the base station with the maximum number of IoT nodes having hard to exploit vulnerabilities. We compute the MTTC before and after reconfiguration of IoT network in each step.

We calculate the MTTC in hours of IoT network from a node, path, and network-level as follows:

We consider $MTTC_{s_n}$ as the mean time to compromise for node level. For example, we calculate the MTTC for 87th node. We calculate the $MTTC_{s_{87}}$ in hours, as follows in Equation 1.

$$\begin{aligned}
MTTC_{s_{87}} &= 1/s_{87_{vul.value}} \\
&= 1/0.71 \\
&= 1.4084 \text{ hours}
\end{aligned} \tag{1}$$

We consider ap as attack path captured in HARM. We consider that there is only one attack path for initial IoT network in HARM. We calculate the $MTTC_{ini}$ as follows in Equation 2.

$$\begin{aligned}
MTTC_{ini} &= \sum_{i=1}^{87} MTTC \\
&= 1.39 + 2.11 + \dots + 2.30 \\
&= 13.79 \text{ hours}
\end{aligned} \tag{2}$$

To explain the MTTC calculation we consider one attack path for initial network. The Equation 3 gives the MTTC value as follows:

$$MTTC_{net} = MTTC_{ini} = 13.79 \text{ hours} \quad (3)$$

We plot the results for increasing number of IoT nodes and the MTTC value in Figure 3.

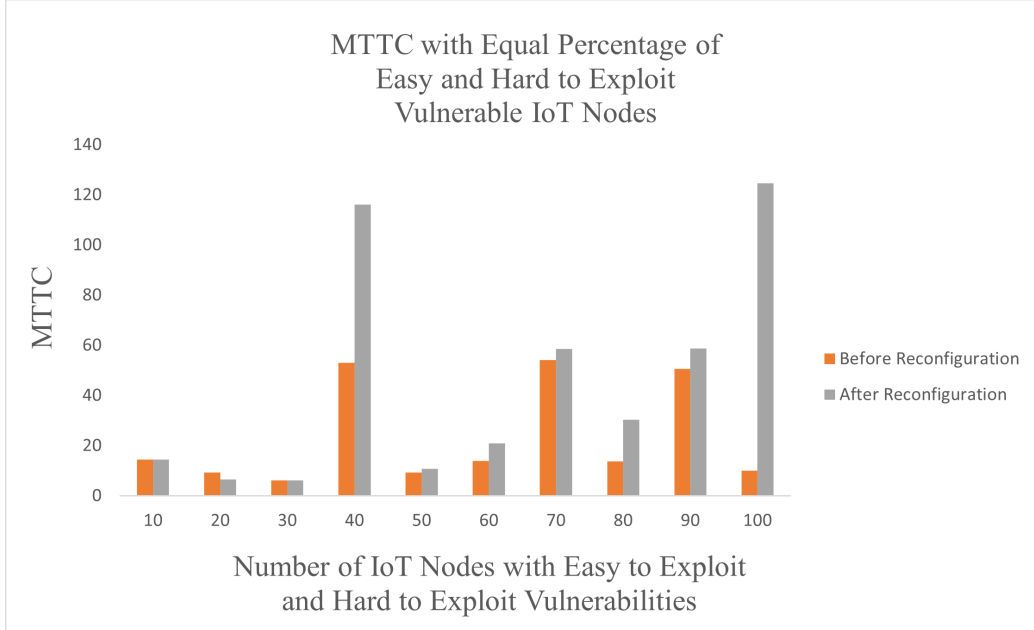


Figure 3: Mean Values of Metric

3.8 Results Discussion

The graph represents the MTTC values against the number of IoT nodes. As explained above, we compute the MTTC before and after IoT network topology reconfiguration. The MTTC increases or decreases based on the attackers entry point node location in the IoT network (as they choose entry point randomly each time the network regenerated) and length of the path from the entry point to the base station. Besides, If the entry point is the last node (100th) in the IoT network, the path is considered as the longest path.

We compare the MTTC values between two scenarios namely before and after topology reconfiguration. In Figure 3, the MTTC value for the scenario of 10 and 30 number of IoT nodes is the same. The reason could be easy to exploit vulnerabilities attached to the IoT nodes and the attack path length. The attack path length could be shorter, after reconfiguration, because the node that was under scanning attack by the attacker, became the child of the node close to the base station.

On the contrary, the MTTC value for the scenario, after topology reconfiguration, has explainable difference in 40 and 100 number of IoT nodes. It is double than the before topology reconfiguration, for 40 number of IoT nodes network and four times for the 100 number of IoT nodes network. Figure 3 depicts that the MTTC is almost higher in after topology reconfiguration scenario except for 20 nodes scenario. We conclude from the results that the attacker put more efforts to reach the base station upon topology

reconfiguration.

Advantages of Reconfiguration: In the reconfigured IoT network, the MTTC increases with the increasing number of IoT nodes having hard to exploit vulnerabilities as the algorithm brings the maximum number of IoT nodes having hard to exploit vulnerabilities in the path. The graph shows the attacker needs more effort to compromise hard to exploit vulnerable IoT nodes in that path. Eventually, the attacker has less gain in the reconfigured IoT network topology.

3.9 Conclusion and Future Work

We consider the reactive defense mechanism for SD-IoT network with easy to exploit and hard to exploit vulnerabilities of IoT nodes. We consider the smart office environment for our SD-IoT network. We evaluate our defense mechanism via simulations. We calculated the MTTC for two IoT network topology scenarios. The scenarios are before topology reconfiguration and after topology reconfiguration. We compare the MTTC values for both scenarios and plot results in the graph.

The proposed defense mechanism works under the assumption of 100 percent detection accuracy of IDS. Besides, our defense mechanism works when one IoT node is under scanning attack. In our future work, we will consider the actual detection accuracy of IDS based on real world scenario. Besides, we will consider the case when attacker scan for more than one IoT nodes at a time and use them as an entry point.

Chapter 4

4 Proactive Defense Mechanism

The proactive defense mechanism is an integrated defense mechanism that combines the deception with MTD and shuffles the IoT network topology proactively. It shuffles the IoT network topology every time the defined security failure conditions (fixed or random time threshold) become true. The proposed defense mechanism uses the deception technology. It is a proactive technique and captures the malicious behavior of the attacker by luring the attacker towards the decoy system [35]. The decoy system includes fake IoT nodes while the attacker does not know whether the IoT nodes are real or fake. Once the attacker interact with the fake IoT node, the malicious behavior of the attacker is recorded.

The proposed defense mechanism also uses the MTD. The MTD techniques shuffle the connection between the IoT nodes when the defined security failure conditions as explained in Section 4.3, become true. In our proposed defense mechanism, we consider the fix shuffling (FS) and random shuffling (RS) MTD techniques for three different network scenarios. The network scenarios are as follows.

- Network with MTD only
- Network with real nodes and decoy nodes only

- Network with Decoy Nodes and MTD

Our key contribution to the proposed integrated proactive defense mechanism is as follows.

- We consider the FS-strategy for the three aforementioned network scenarios and compare the results.
- We consider the RS-strategy for these three network scenarios and compare the results.
- We use two new metrics that are APV and APE, to measure the effectiveness of our proposed defense mechanism.

The APV and APE metrics measure the effectiveness of the proposed defense mechanism in terms of a shift in the attack surface. These metrics offer new attack paths at a defined time interval to make the attack surface harden. In addition to the APV and APE metrics, we also measured the mean-time-to-security-failure (MTTSF) metric and Defense Cost (DC) metric proposed by [36].

4.1 Scenario Description

We propose a proactive integrated defense mechanism. The main goal of our defense mechanism is to secure the SD-IoT network using a decoy system as cyber-deception and topology reconfiguration as an MTD technique.

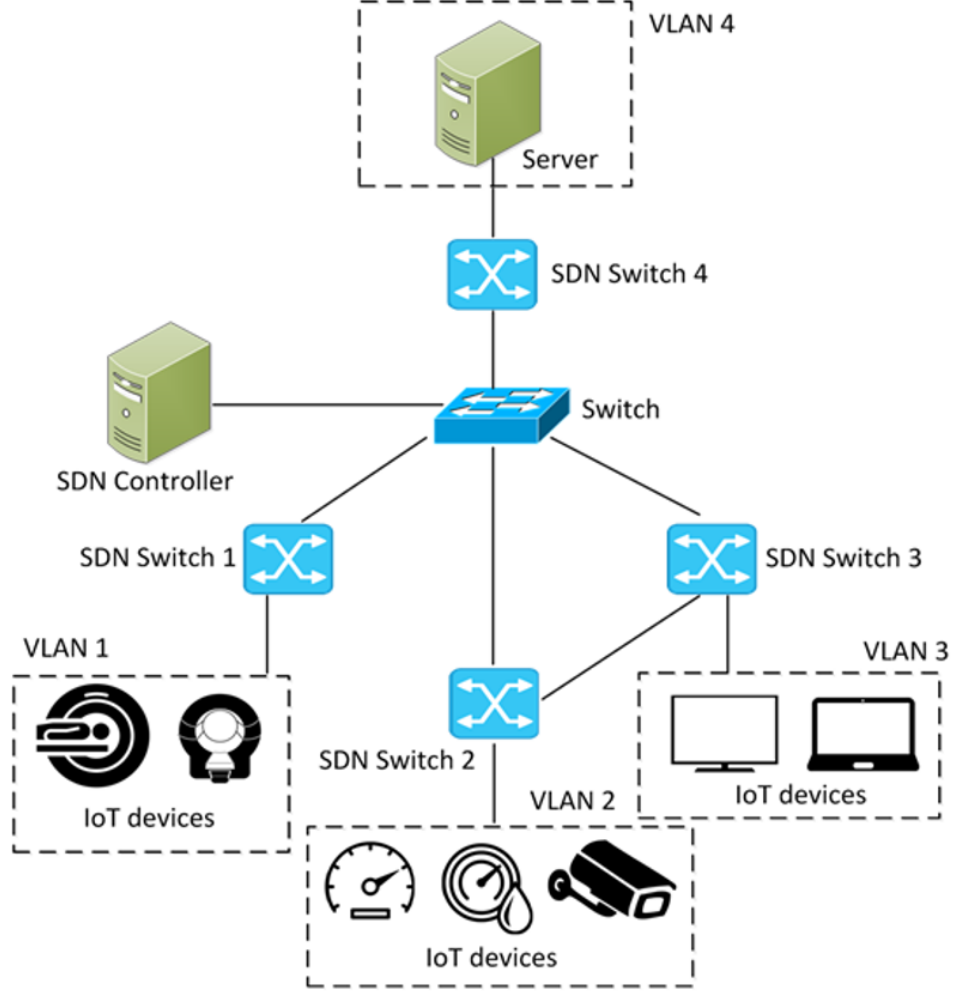


Figure 4: SD-IoT Network [36]

To evaluate our integrated defense mechanism, we use HARM which gives the design solution by considering the attack graphs and then calculates the security level for all attack paths. We use the example SD-IoT network as shown in Figure 4 proposed by [36], for our proactive defense mechanism.

The SD-IoT network consists of an SDN controller, SDN switches, servers,

and IoT nodes. The IoT nodes and the servers are placed in different VLANs, based on the different functionalities. The IoT nodes are a mix of real nodes and decoy nodes. We consider the SDN programmability to manage and control the communication between IoT nodes.

4.2 Methodology

Our methodology consists of a network model, attack model, and defense model. The network model describes the SD-IoT network, the attack model explains the assumptions we make for the attacker attached with the IoT network and the defense model explains the defense strategies.

4.2.1 Network Model

Our network model is an SD-IoT network in a smart hospital environment. The IoT network consists of different IoT nodes and are used for different purposes. The IoT nodes are namely, MRI, CT Scan, Smart Thermostat, Smart Meter, Smart Camera, Smart TV, Laptop, and a server [37]. These IoT nodes are placed in different VLANs as shown in Figure 4. The SDN controller manages and controls the IoT nodes via SDN switches and IoT nodes send the collected data to the server for further query processing. In our SD-IoT network, we consider some of the nodes are compromised and some of the nodes have critical information. We also deploy decoy nodes in each VLAN of the IoT network. Finally, we specify the vulnerabilities attached to the IoT nodes.

4.2.2 Attack Model

We consider the assumptions for an attacker attached to the SD-IoT network and assign the privileges to the attacker as follows.

- We assume that the attacker has less knowledge about the IoT node whether it is a real node or decoy node. The attacker's knowledge depends on how much they get information about the IoT network, on interaction with the IoT network.
- On attacker's interaction with the decoy, the intentions of the attacker are recorded. However, if the attacker knows that the node with which they are interacting is a decoy node, the attacker terminates the interaction and find another node to break into the system.
- We assume that the main target of the attacker is to leak confidential information to an unauthorized party outside the SD-IoT network.
- We also consider that the attacker can identify the unpatchable vulnerabilities and can compromise these vulnerabilities to break into the SD-IoT network.
- The attacker can only compromise vulnerable IoT nodes and then can reach the server, which is the main target.

The vulnerabilities attached to the IoT nodes are collected from Common CVE/NVD [38]. In our assumptions, we consider that there is one exploitable

vulnerability attached to each IoT node and attackers exploit this vulnerability to break into the system. We also consider the compromise rate for the vulnerabilities shown in Table 2. It shows the attacker's frequency to exploit the vulnerability successfully and to gain the root privilege per unit time (hour). The compromise rate value is an estimation based on the score from the CVSS [38]. We calculate the compromise rate value depends on the base score as follows.

- If the base score is 10.0, we estimate the compromise rate once per day (i.e., 0.042).
- If the base score is 8.0, we estimate the compromise rate twice per week (i.e., 0.012).
- If the base score is around 7.0, we estimate the compromise rate once per week (i.e., 0.006).
- If the base score is around 5.0, we estimate the compromise rate once per 10 days (i.e., 0.004)

Table 2 shows the real node vulnerabilities, their detail, and the compromise rate. Similarly, Table 3 shows the decoy node vulnerabilities, their detail, and the compromise rate.

Table 2: Real Node and Vulnerability Information

Real Node	VLAN	CVE ID	Compromise Rate
MRI	VLAN1	CVE-2018-8308	0.006
CT Scan	VLAN1	CVE-2018-8308	0.006
Smart Thermostat	VLAN2	CVE-2018-11315	0.006
Smart Meter	VLAN2	CVE-2017-9944	0.042
Smart Camera	VLAN2	CVE-2018-10660	0.042
Smart TV	VLAN3	CVE-2018-4094	0.012
Laptop	VLAN3	CVE-2018-8345	0.004
Server	VLAN4	CVE-2018-8273	0.006

Table 3: Decoy Node and Vulnerability Information

Decoy Node	VLAN	CVE ID	Compromise Rate
CT Scan	VLAN1	CVE-2018-8136	0.012
Smart Camera	VLAN2	CVE-2018-6294	0.042
Smart TV	VLAN3	CVE-2018-4095	0.012
Server	VLAN4	CVE-2018-1930	0.042

4.2.3 Defense Model

We assume that the SD-IoT network has its traditional defense mechanisms which consist of Network based-IDS, anti-virus software on server. However, we add the intrusion prevention mechanism on top of the traditional defense mechanisms and integrate the cyberdeception with MTD.

The cyberdeception technique attracts the attacker by offering exploitable vulnerabilities. The attacker takes the advantage of exploitable vulnerability and interact with it to break into the IoT network. Once the attacker interacts with the IoT nodes, the decoy system monitors the behavior of the attacker and reveals the intentions of the attacker to the defender. The IoT

nodes of the decoy system are mimic of real IoT nodes and it is difficult for the attacker to identify whether the IoT node is real or decoy.

We use network topology shuffling as an MTD defense mechanism (NTS-MTD). NTS-MTD triggers when the intrusion event is detected in the SD-IoT network. The SDN controller manages and changes the connections between the IoT nodes on topology reconfiguration. Each VLAN has a decoy IoT node and the connection changes take place from real IoT node to decoy IoT node, decoy IoT nodes to decoy IoT node and decoy IoT node to decoy server. After reconfiguration, the resulted network topology consists of randomly distributed real and decoy nodes as the connections are changed and makes the attack surface complex for the attacker.

4.3 Security Failure Conditions

We consider that the attacker can enter into the IoT network by performing the reconnaissance attack. The reconnaissance attack causes the failure of the IoT network integrity and allows the outside attacker to penetrate the network using different scanning techniques. Once the attacker identify the vulnerable IoT node in the IoT network, he/she try to compromise that node and break into the IoT network. Now the attacker can perform the data exfiltration attack. This attack causes the loss of confidentiality of the IoT network. Now the attacker can use different credentials to compromise the nodes with confidential information.

Based on these two attacks, we define two security failure conditions.

We consider our IoT network compromised if either of the security failure condition becomes true.

Security Failure Condition 1 (SFC1): SFC1 occurs because of the loss of integrity. It uses the concept of Byzantine Failure [39], that is one-third of the nodes in the network, are compromised.

Security Failure Condition 2 (SFC2): SFC2 occurs because of the loss of confidentiality. The confidential information (e.g., login details, passwords etc) is leaked to outside unauthorized entities by the attacker who compromise the IoT nodes.

4.4 Metrics

To measure the security and performance of our proposed defense mechanism, we use the following metrics.

- **Attack Path Variation (APV):** It measures the change in attack paths upon SD-IoT network topology reconfiguration.
- **Attack Path Exposure (APE):** This metric measures the time duration (the time a path is staying unchanged in the network) of attack paths appeared in one state of network.
- **Mean Time to Security Failure (MTTSF):** This metric measures the network lifetime indicating how long the network prolongs until the network reaches the security failure.

- **Defense Cost (DC):** It gives the cost associated with shuffling operations. we count the number of edges shuffled (i.e., from connected to disconnected and from disconnected to connected).

4.4.1 Calculation of Metrics

APV: The APV measures the shift in the attack path upon reconfiguration of IoT network topology. As a result, the new attack paths appear and increase the attack effort. It is because, when the attack path is changed, the attacker loses the gathered information about the intended attack path and start to redesign the attack strategy.

To calculate the APV, we consider the attack paths difference, $\Delta AP_{AP_i, AP_{i-1}}$ before and after reconfiguration (between two network states, i and i-1). Here, the AP_i gives the set of attack paths in i^{th} state of the network. The variation in the current set of attack paths is the difference from the previous set of attack paths (Set Subtraction). For example in network state 1 , we have the set of attack paths as $\{a \rightarrow b \rightarrow c, a \rightarrow d \rightarrow c, a \rightarrow e \rightarrow c\}$. In-network state 2, we apply MTD mechanism and get the set of attack paths as $\{a \rightarrow b \rightarrow c, a \rightarrow f \rightarrow c\}$.

Here “d” changed to “f” that represents a new attack path and the attack path $\{a \rightarrow e \rightarrow c\}$ does not appear upon reconfiguration.

Let “ - ” define as to subtract the two set. Now the difference in the set of attack paths is as shown in Equation 4.

$$\{a \rightarrow b \rightarrow c, a \rightarrow f \rightarrow c\} - \{a \rightarrow b \rightarrow c, a \rightarrow d \rightarrow c, a \rightarrow e \rightarrow c\} = \{a \rightarrow f \rightarrow c\} \quad (4)$$

The set of difference from the previous network state to the current one reveals the number of new attack paths which were not in the previous network state. In the above example, it is $\{a \rightarrow f \rightarrow c\}$.

To reveal the attack paths difference for i number of network states, we consider the following equation.

$$\Delta AP_{i,i-1} = \frac{|AP_i - AP_{i-1}|}{|AP_i|} \quad (5)$$

Where $|AP_i - AP_{i-1}|$ is the cardinal value which represents the count of difference in attack paths between two states.

To compute the APV metric for all the network states “S” on reconfiguration, we use the following equation.

$$APV = \frac{\sum_{i=1}^{|S|} \Delta AP_{i,i-1}}{|S| - 1} \quad (6)$$

Here, $|S|$ is cardinal value that is the attack paths in one state.

The above equation represents the overall changes in the attack paths and their variations in all the observed network states “S”.

We develop Algorithm 2 to calculate the APV. We use the following general notations to explain the algorithm.

- b_i : The IoT nodes in the network ($i \in \{1, \dots, n\}$). Where n is the number of IoT nodes.
- h_i : Single attack path in a list of attack paths ($i \in \{1, \dots, x\}$). Where x is the number of attack paths.
- n_{i-1} : Previous network state.
- m_i : Current network state.
- K : List of new attack paths.
- C : A list of all the attack paths in the network.
- Q : The attack paths in current network state.
- R : The attack paths in previous network state.
- P : Set of attack paths appeared, after reconfiguration.
- L : Adding the time duration (attack path appearance time) randomly for each attack path.
- U : Total time duration of the attack path appeared in one network state.
- A^* : To initialize variables.
- “ / ”: Operator for division.
- “ - ”: To subtract the values.

- “ * ”: To multiply the two values.

Algorithm 2: Attack Path Variation (APV)

Result: Attack path variation solution

```

1 for each attack path  $\in m$  and  $n$  do
2    $n_{i-1} = R$ ;
3    $m_i = Q$ ;
4    $ap \leftarrow C$ ;
5   for  $b_i \in m_i$  do
6     if  $b_i$  is not in  $n_{i-1}$  then
7       Add  $b_i$  into  $m_i$ 
8     end
9      $K = P/m_i$ ;
10    Add  $K$  into Result ;
11  end
12 end
13 Return (Result/ $C - 1$ )

```

The explanation of the Algorithm 2 is as follows.

In the algorithm we go through all the attack paths in the current network state and the previous network state (line 1). We get the attack paths in the previous network state and current network state (line 2,3). We initialize the attack path list (line 4). We go through each node b_i in the current network state (line 5), we check if any of the node which was not in the previous network state (line 6). We add that node into current network state (line 7). The addition of new node represents the appearance of new attack path and we get the list of new attack paths appeared in the current state (line 9). We add these new attack paths into the attack path variation solution in line 10. Finally, we recover the attack path variation solution by dividing

the result with the total number of attack paths in the network.

APE: This metric gives the time duration of APE for attacker [40]. The attack can be launched successfully if the attack path exposure time is long and it is desired to keep it minimum. We use the following equation to calculate the APE.

$$APV = 1 - \frac{\sum_{i=0}^{|S|} t(ap_j)}{|AP| * \sigma_{i=1}^{|S|} t(s_i)} \quad (7)$$

where $t(s_i)$ represents the time duration (in hour) of i^{th} network state, $t(ap_j)$ represents the time duration of an attack, exploiting the attack path ap_j (single attack path) and AP represents the set of attack paths. Where ap_j is the attack path out of the set of attack paths in AP_i ($\forall ap_j \in AP_i$).

If the initial attack paths are exposed to all the network states without any new attack paths, then the APE value tends toward zero. The ideal condition is that the attack path appears in the new network state, upon reconfiguration, may not appear in the initial network state. It means the exposure duration of that specific attack path is minimum.

We develop Algorithm 3 to calculate the APE and use the general notations, given above, to explain the algorithm. The explanation of the Algorithm 3 is as follows.

We initialize the attack path exposure solution, total time duration of an attack path in one network state and the list of new attack paths (line 1-4) respectively. We go through all the attack paths in the attack path list (line 5). We get the attack paths in the current network state m_i at line 6. We

add the time to all the attack paths in the current state of network (line 7). We go through all the attack paths again and check for each attack path if it is not in the new attack path list K (line 8,9) and add that attack path to the new attack path list (line 10). We add the new attack path duration time into total time and return the attack path exposure solution (line 12-15).

Algorithm 3: Attack Path Exposure (APE)

Result: Attack path exposure solution

```

1 Initialization;
2  $L \leftarrow A^*$ ;
3  $U \leftarrow A^*$ ;
4  $K \leftarrow A^*$ ;
5 for each attack path  $\in K$  do
6    $m_i = Q$ ;
7   Add  $m_i * K$  into L;
8   for  $h_i \in m_i$  do
9     if  $h_i$  is not in  $K$  then
10      Add  $h_i$  into K;
11     end
12   Add L into U;
13 end
14 end
15 Return  $1 - (L / K * U)$ 

```

4.5 Simulation

We run the simulations 100 times for each of the network scenario. In simulation, we explain the simulation settings, schemes used for the shuffling of topology, and performance analysis. We use the desktop machine with the operating system, Windows 10 64-bit, 8.0 GB RAM, and Core i5-8250u

(1.60GHz) processor. We implement our proposed defense mechanism based on the workflow as shown in Figure 5

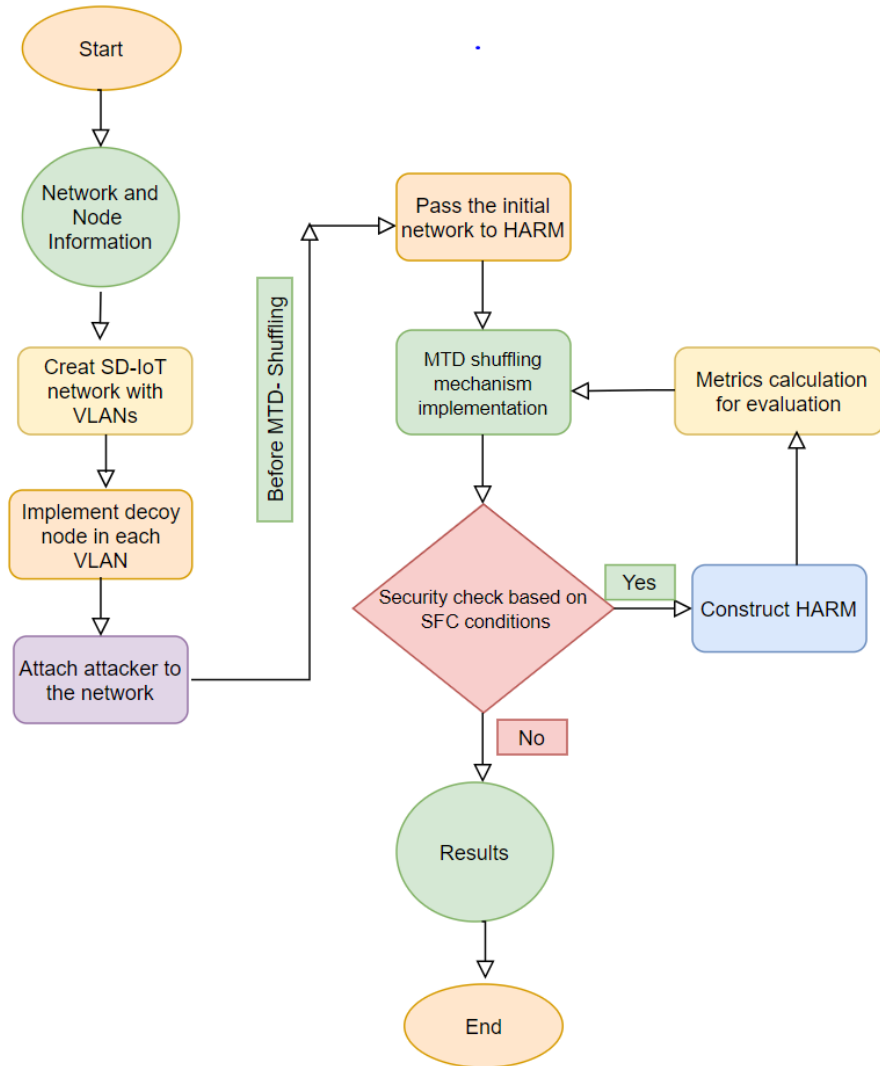


Figure 5: Workflow of Defense Mechanisms

4.5.1 Simulation Setting

We consider the smart hospital environment equipped with IoT nodes in our simulation setting as shown in Figure 4. The IoT nodes are deployed in different VLANs. Our IoT network consists of four VLANs.

In VLAN1, we deploy an MRI and CT Scan and consider as a medical examination room. In VLAN2, we deploy a smart thermostat, smart meter, and a smart camera, and consider as a medical care unit. In VLAN3, we deploy smart TV and a laptop and consider it as a staff office. We consider the VLAN4 as a server room.

During the initial deployment, the VLAN4 is connected with all the three VLANs. It is because the IoT nodes in the other three VLANs send the gathered information to the server placed in VLAN4, for further query processing. The VLAN3 is a staff room and it is connected with VLAN2, to receive the medical data and videos from the smart camera and to monitor and control the sensors. All the IoT nodes have vulnerabilities and attacker can exploit these vulnerabilities to compromise the IoT nodes. The vulnerabilities for the real nodes are given in Table 2.

For cyberdeception, we deploy decoy nodes on VLANs, and in our network scenario, we deploy one decoy node in each VLAN. There are two types of decoy nodes we use in VLANs.

- **Emulated decoy node:** It is a fake asset node that is used to deceive the attacker. The fake nodes are the mimic of the original nodes with

the same functionality as the real nodes.

- **Full-Operating System (OS) decoy node:** The full operating system decoy nodes are the replication of the real operating system and software on the production devices such as server..

We consider a Full-OS-based server as a decoy and emulated based decoy nodes which are CT Scan, Smart TV, and Smart camera. The decoy nodes are also attached to the vulnerabilities and attackers exploit these vulnerabilities to gain the root access of the decoy nodes. The vulnerabilities of the decoy nodes are given in Table 3.

4.5.2 Network Scenarios to Calculate Metrics

To calculate the metrics for performance analysis, we consider three network scenarios. We name these scenarios as "How-to-shuffle" network topology.

Scenario with MTD Only: In this scenario, we consider the SD-IoT network without decoy nodes and we only consider the MTD technique to shuffle the network topology. This scenario uses a random shuffling algorithm. We calculate the proposed metrics for fixed time interval topology shuffling and random time interval topology shuffling.

Scenario with Decoy Nodes Only: In this scenario, we consider the SD-IoT network with real nodes and decoy nodes. we randomly add the connections between real nodes, from real nodes to decoy nodes and decoy nodes to decoy nodes. We do not add the connections from decoy nodes

to real nodes. The flow is one way. The decoy nodes always communicate with decoy nodes, and the attacker can not moves to real nodes if they are interacting with decoy nodes. Finally, We apply fixed time interval topology shuffling and random time interval topology shuffling to reconfigure the topology and to calculate the metrics for evaluation.

Scenario With Decoy Nodes and MTD: For this scenario, we consider a network with decoy nodes along with the MTD technique. We use the heuristic shuffling algorithm [36] and shuffle the connections between real nodes to decoy nodes to give the maximum number of decoy attack paths in the SD-IoT network. The algorithm selects the optimal network topology which offers the maximum number of decoy attack paths to the attacker. Finally, we calculate the metrics for a fixed time interval and random time interval techniques.

4.5.3 Six schemes to compare the metric results based on when-to-shuffle and how-to-shuffle strategies

We consider two when-to-shuffle network topology strategies as follows.

FS: This strategy shuffles the network topology at fix time interval.

RS: This strategy shuffles the network topology at a random time interval. The random time interval is achieved by following the variation in fix strategy. The fix strategy is the distribution with the mean being the same as the fixed time interval. This will add some stochastic nature to the time interval which is treated as a random variable.

Table 4: Keywords, their meanings and default values.

Keyword	Meaning	Value
SFC1	Loss of system integrity	0.5
SFC2	Loss of system confidentiality	0.5
Prob.	Probability of change in connection between nodes in RS	0.5
Time	FS time interval used (hour)	24
Path time	Attack path exposure time in each network state (hour)	1-5
Mean time	Mean time used for exponential distribution in RS (hour)	24

By combining when-to-shuffle strategies with “how-to-shuffle” network topology, we get six schemes as FS-MTD, FS-Decoy, FS-Decoy-MTD, RS-MTD, RS-Decoy, and RS-Decoy-MTD. We perform the metric values comparative analysis for these six schemes and conclude which scheme outperforms.

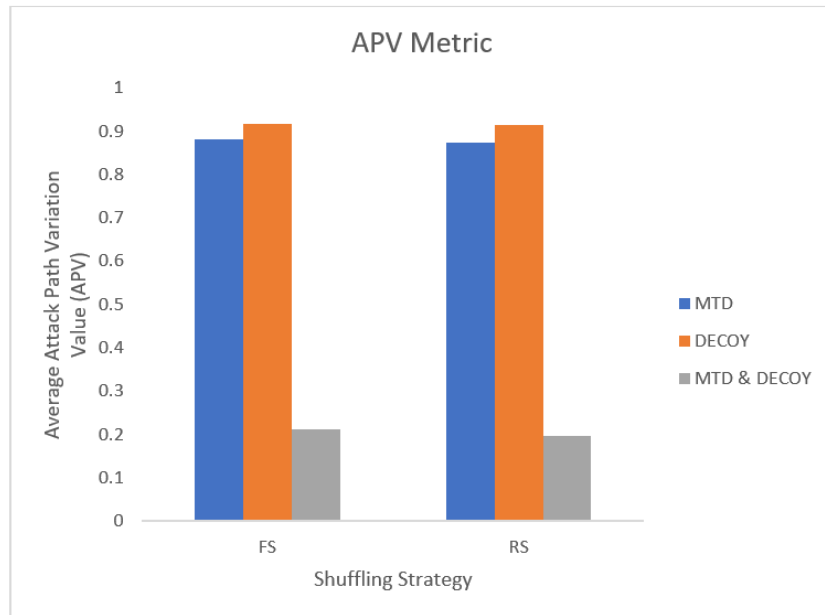
Table 4 presents the keywords, their meaning and default value which we use in our simulation.

4.5.4 Comparative Analysis

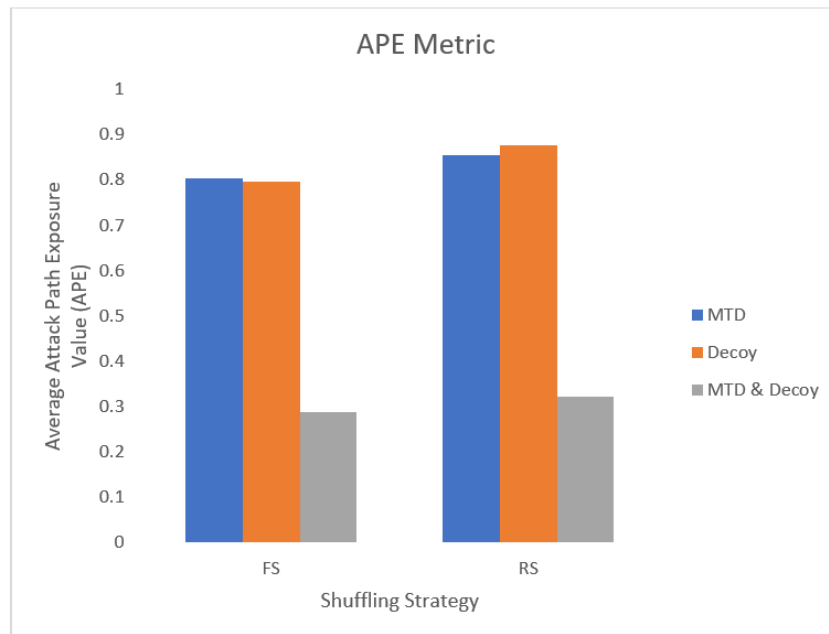
We carried out the performance analysis by comparing the six schemes as mentioned above. We use the Fix shuffling and random shuffling strategies for each scenario to calculate the APV, APE, MTTSF, and DC.

Figure 6 gives the performance analysis comparison for 6 schemes.

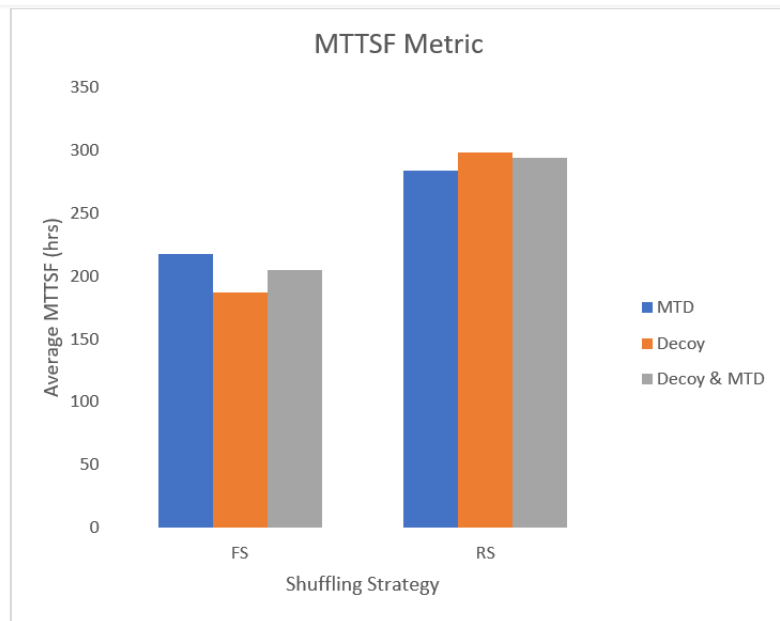
- Figure 6a compares the APV for three network scenarios. The higher APV value represents an effective solution. It means the attack path that appeared in one state does not appear in the second state of the



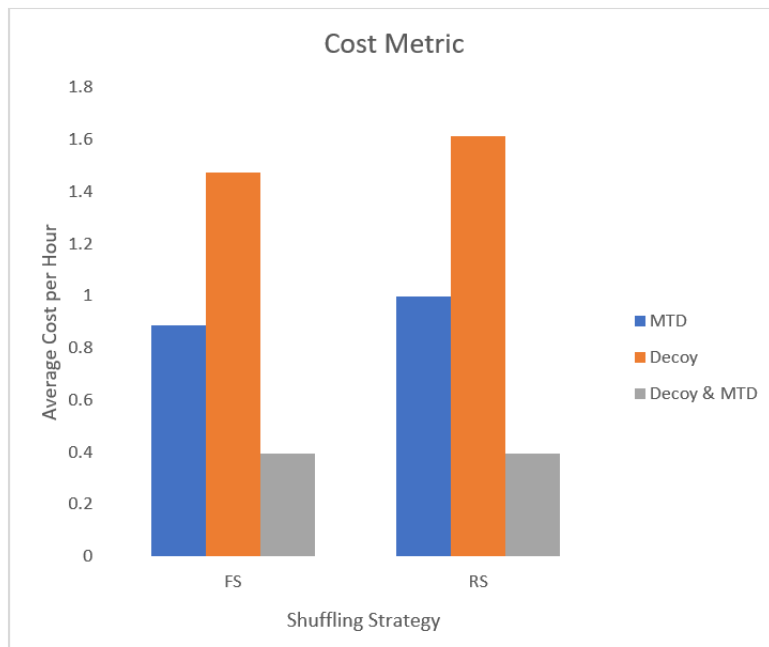
a) Shuffling Strategy vs APV



b) Shuffling Strategy vs APE



c) Shuffling Strategy vs MTTSF



d) Shuffling Strategy vs CD

Figure 6: Shuffling Strategies Comparisons

network. In the “when-to-shuffle” strategy, the FS performs better than the RS strategy. In the “how-to-shuffle” strategy, the only decoy based network scenario scheme performs comparatively better than the other two. Overall, the “MTD & Decoy Network Scenario” has less value than the MTD only and Decoy only network scenario. It is because the topology change takes place less number of times in “MTD and Decoy network scenario” than the “MTD only network scenario” and “Decoy only network scenario”. If the topology change takes place more number of times, the more new attack paths appear and APV value becomes high.

- Figure 6b compares the APE for “how-to-shuffle” network topology scenarios. The lower APE value represents the effective solution. It means the attack path appearance duration is minimum and the attack path variation is high. The high attack path variation shows that the attack path stays for less time and replaced by new attack path. In the “when-to-shuffle” strategy, the FS strategy performs better. For the “how-to-shuffle” network topology scenario, the MTD & Decoy-based network scenario for fixed time intervals performs better. The reason behind this is the topology reconfiguration takes place in a fix time interval for FS MTD & Decoy-based network scenario, even if the intrusion is not detected. At the fix time interval the topology reconfigures and new attack paths appear that represent the high attack path variation and low APE value. In addition, the MTD & Decoy

based network scenario gives more new connections from real node to decoy nodes and gives more attack paths on reconfiguration. It results the less APE value.

- Figure 6c compares the average MTTSF for three network scenarios. The higher value of MTTSF represents the effective solution which means the attacker takes more time to compromise the network. In the “when-to-shuffle” strategy, the RS performs better than the FS. On the other hand, in the “how-to-shuffle” strategy, the MTTSF is higher for decoy based network scenario. It means the attacker gets the higher deception level and follows the path leads to the decoy server.

Overall, the RS strategy performs better than the FS strategy to calculate the MTTSF.

The RS strategy reconfigures the topology at a random time interval. In result, it reconfigures the topology more often than FS strategy and attacker loses the information upon reconfiguration and try to find another entry point to launch the attack.

- Figure 6d compares the average DC for three network scenarios. The lower value of DC means an effective solution. The DC is associated with the number of connections change, each time the network shuffling takes place. In the “when-to-shuffle” strategy, the FS strategy performs slightly better than the RS strategy. On the other hand, the DC for the decoy and MTD network scenario is lower and it outperforms than

other network scenarios.

In the MTD only network scenario and Decoy only network scenario, the security failure conditions might be achieved earlier than the Decoy and MTD network scenario. When the threshold is achieved, topology reconfiguration and connection changes between the nodes take place. The new attack paths appear upon connection changes. The number of attack paths are inversely proportional to the DC. It leads to a higher value of DC in MTD only network scenario and Decoy only network scenario while lower value of DC in Decoy and MTD network scenario.

4.6 Results Discussion and Conclusion

From the results, we can see that the deployment of decoy nodes in the IoT network affects the attacker's abilities to compromise the IoT nodes. However, it gives higher defense costs among the other network scenarios. There is no single scheme that achieves all the goals of maximizing APV (Figure 6a), MTTSF (Figure 6c), and minimizing the APE (Figure 6b), DC (Figure 6d). The graphs represent that the decoy network scenario is the best "how-to-shuffle" strategy for APV and MTTSF while the decoy and MTD scenario is best for APE and DC. In conclusion, if we do not consider the DC factor, then a network scenario with decoy nodes is the optimal solution for our SD-IoT network.

Chapter 5

5 Conclusion

The thesis addresses all the research questions we proposed in Chapter 1 by developing the reactive defense mechanism for the SD-IoT network and developing the integrated proactive defense mechanism for the SD-IoT network.

To address the research question Q1 in Section 1.2, We have proposed an integrated proactive defense mechanism for the SD-IoT network. We used deception technology with MTD as a defense strategy. We considered the smart hospital environment as an SD-IoT network. We used VLANs in the IoT network and deployed the IoT nodes in different VLANs according to their functionality. Besides, each VLAN also has a decoy node in it. To evaluate our defense mechanism, we measure the security metrics that are APV, APE, MTTSF, and DC.

To address the research question Q2 in Section 1.2, we have proposed the reactive defense mechanism for SD-IoT network. We consider the easy to exploit and hard to exploit vulnerabilities attached with nodes in SD-IoT network. We considered the smart environment as an IoT network and performed simulations. We designed and implemented the reconfiguration algorithm, and used varying percentage proportions of easy to exploit and hard to exploit IoT nodes in our simulation. We measure the MTTC as a

security metric to present the effectiveness of our defense mechanism.

5.1 Limitations and Future Work

In the reactive defense mechanism, we assume it works under the assumption of hundred percent detection accuracy of IDS. Besides, our defense mechanism works when one IoT node is under attack. The limitations can be investigated in future research such as the attacker can scan for more than one IoT nodes at a time and use them as an entry point.

In the proactive defense mechanism, we consider the small scale smart hospital environment. Besides, we also assume that the attacker is not aware of the decoy system in the SD-IoT network. In our future work, we will develop a large-scale SD-IoT network with distributed MTD techniques. For the decoy system, we will introduce the attacker's intelligence about the SD-IoT network. We will define the levels of attacker's intelligence and will measure the harm caused by the attacker at different intelligence levels.

References

- [1] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, Privacy and Trust in Internet of Things: The Road Ahead,” *Computer Networks*, vol. 76, pp. 146–164, 2015.
- [2] R. Roman, P. Najera, and J. Lopez, “Securing the Internet of Things,” *IEEE Computer*, vol. 44, no. 9, pp. 51–58, Sep. 2011, ISSN: 0018-9162. DOI: 10.1109/MC.2011.291.
- [3] P. McDermott-Wells, “Bluetooth scatternet models,” *IEEE potentials*, vol. 23, no. 5, pp. 36–39, 2004.
- [4] P. S. Henry and H. Luo, “Wifi: What’s next?” *IEEE Communications Magazine*, vol. 40, no. 12, pp. 66–72, 2002.
- [5] P. Kinney *et al.*, “Zigbee technology: Wireless control that simply works,” in *Communications design conference*, vol. 2, 2003, pp. 1–7.
- [6] Y. DA, L. ZHANG, and K. YANG, “A DDoS Attack Detection and Mitigation with Software-Defined Internet of Things Framework. IEEE Access, 2018,” *Computer Networks*,
- [7] M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei, “Emergent properties: Detection of the node-capture attack in mobile wireless sensor networks,” in *Proceedings of the first ACM conference on Wireless network security*, 2008, pp. 214–219.

- [8] J. Deogirikar and A. Vidhate, “Security attacks in iot: A survey,” in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, IEEE, 2017, pp. 32–37.
- [9] K. Chung, C. A. Kamhoua, K. A. Kwiat, Z. T. Kalbarczyk, and R. K. Iyer, “Game Theory with Learning for Cyber Security Monitoring,” in *the Proceedings of the 17th IEEE International Symposium on High Assurance Systems Engineering (HASE), 2016*, IEEE, 2016, pp. 1–8.
- [10] J. Voas, R. Kuhn, C. Kolias, A. Stavrou, and G. Kambourakis, “Cybertrust in the IoT Age,” *IEEE Computer*, vol. 51, no. 7, pp. 12–15, 2018.
- [11] Q. Lv, “Research on state monitoring Technology for Power Equipment Based on Internet of Things,” in *the Proceedings of the International Conference on Robots & Intelligent System (ICRIS) 2018*, IEEE, 2018.
- [12] J. Hong and D.-S. Kim, “Harms: Hierarchical attack representation models for network security analysis,” 2012.
- [13] H. Kim and N. Feamster, “Improving network management with software defined networking,” *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.
- [14] S. Yoon, T. Ha, S. Kim, and H. Lim, “Scalable traffic sampling using centrality measure on software-defined networks,” *IEEE Communications Magazine*, vol. 55, no. 7, pp. 43–49, 2017.

- [15] J. Liu, Y. Li, M. Chen, W. Dong, and D. Jin, “Software-Defined Internet of Things for Smart Urban Sensing,” *IEEE Communications Magazine*, vol. 53, no. 9, pp. 55–63, 2015.
- [16] A. Hakiri, P. Berthou, A. Gokhale, and S. Abdellatif, “Publish/Subscribe-Enabled Software Defined Networking for Efficient and Scalable IoT Communications,” *IEEE Communications Magazine*, vol. 53, no. 9, pp. 48–54, 2015.
- [17] H. Sándor, B. Genge, and G. Sebestyén-Pál, “Resilience in the Internet of Things: The Software Defined Networking Approach,” in *the Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2015*, IEEE, 2015, pp. 545–552.
- [18] G. Grigoryan, Y. Liu, L. Njilla, C. Kamhoua, and K. Kwiat, “Enabling Cooperative IoT Security via Software Defined Networks (SDN),” *arXiv preprint arXiv:1806.01885*, 2018.
- [19] S. Chakrabarty, D. Engels, and S. Thathapudi, “Black SDN for the Internet of Things,” in *Proceedings of 12th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pages 190–198, 2015..*
- [20] N. Kushalnagar, G. Montenegro, C. Schumacher, *et al.*, “Ipv6 over low-power wireless personal area networks (6lowpans): Overview, assumptions, problem statement, and goals,” 2007.

- [21] T. Miyazaki, S. Yamaguchi, K. Kobayashi, J. Kitamichi, S. Guo, T. Tsukahara, and T. Hayashi, “A software defined wireless sensor network,” in *2014 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2014, pp. 847–852.
- [22] Q. D. La, T. Q. Quek, J. Lee, S. Jin, and H. Zhu, “Deceptive attack and defense game in honeypot-enabled networks for the internet of things,” *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1025–1035, 2016.
- [23] M. Anirudh, S. A. Thileeban, and D. J. Nallathambi, “Use of honeypots for mitigating dos attacks targeted on iot networks,” in *Proceedings of the IEEE International Conference on Computer, Communication and Signal processing (ICCCSP, pages 1–4, 2017)*.
- [24] R. Zhuang, S. A. DeLoach, and X. Ou, “Towards a theory of moving target defense,” in *Proceedings of the First ACM Workshop on Moving Target Defense*, 2014, pp. 31–40.
- [25] J. B. Hong and D. S. Kim, “Assessing the effectiveness of moving target defenses using security models,” *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 163–177, 2015.
- [26] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, “Toward proactive, adaptive defense: A survey on moving target defense,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

- [27] M. Ge, J. B. Hong, W. Guttman, and D. S. Kim, “A framework for automating security analysis of the internet of things,” *Journal of Network and Computer Applications*, vol. 83, pp. 12–27, 2017.
- [28] M. Sherburne, R. Marchany, and J. Tront, “Implementing moving target ipv6 defense to secure 6lowpan in the internet of things and smart grid,” in *Proceedings of the 9th Annual Cyber and Information Security Research Conference*, 2014, pp. 37–40.
- [29] K. Zeitz, M. Cantrell, R. Marchany, and J. Tront, “Designing a micro-moving target ipv6 defense for the internet of things,” in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, IEEE, 2017, pp. 179–184.
- [30] M. Ge, J. B. Hong, S. E. Yusuf, and D. S. Kim, “Proactive defense mechanisms for the software-defined internet of things with non-patchable vulnerabilities,” *Future Generation Computer Systems*, vol. 78, pp. 568–582, 2018.
- [31] A. I. Kouachi, S. Sahraoui, and A. Bachir, “Per packet flow anonymization in 6lowpan iot networks,” in *2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, IEEE, 2018, pp. 1–7.
- [32] F. Nizzi, T. Pecorella, F. Esposito, L. Pierucci, and R. Fantacci, “Iot security via address shuffling: The easy way,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3764–3774, 2019.

- [33] B. Caswell, J. Beale, and A. Baker, *Snort intrusion detection and prevention toolkit*. Syngress, 2007.
- [34] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, “Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, IEEE, 2015, pp. 513–521.
- [35] L. Pingree, “Emerging technology analysis: Deception techniques and technologies create security technology business opportunities,” *Gartner, Inc*, 2015.
- [36] M. Ge, J.-H. Cho, D. S. Kim, G. Dixit, and I.-R. Chen, “Proactive defense for internet-of-things: Integrating moving target defense with cyberdeception,” *arXiv preprint arXiv:2005.04220*, 2020.
- [37] S. Li, L. Da Xu, and S. Zhao, “The internet of things: A survey,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.
- [38] S. Radack, “National vulnerability database (nvd): Helping information technology system users and developers find current information about cyber security vulnerabilities,” National Institute of Standards and Technology, Tech. Rep., 2005.
- [39] F. C. Gärtner, “Byzantine failures and security: Arbitrary is not (always) random,” *INFORMATIK 2003-Mit Sicherheit Informatik, Schwerpunkt” Sicherheit-Schutz und Zuverlässigkeit*, 2003.

- [40] J. B. Hong, S. Y. Enoch, D. S. Kim, A. Nhlabatsi, N. Fetais, and K. M. Khan, “Dynamic security metrics for measuring the effectiveness of moving target defense techniques,” *Computers & Security*, vol. 79, pp. 33–52, 2018.